

# Algorithms for cliques in inductive $k$ -independent graphs

George Manoussakis<sup>1</sup>

<sup>1</sup>LRI-CNRS, University Paris Sud, France.

## Abstract

A graph is inductive  $k$ -independent if there exists an ordering of its vertices  $v_1, \dots, v_n$  such that  $\alpha(G[N(v_i) \cap V_i]) \leq k$  where  $N(v_i)$  is the neighbourhood of  $v_i$ ,  $V_i = \{v_i, \dots, v_n\}$  and  $\alpha$  is the independence number. In this article we design a polynomial time approximation algorithm with ratio  $\bar{\Delta} / \log(\log(\bar{\Delta}) / (k+1))$  for the maximum clique problem and also show that the decision version of this problem is fixed parameter tractable, with parameter  $p$ , the size of the clique, for this particular family of graphs, with complexity  $\mathcal{O}(p^2(p+k-1)^{kp}n)$ . Then we study a subclass of inductive  $k$ -independent graphs, namely  $k$ -degenerate graphs. A graph is  $k$ -degenerate if any induced subgraph has a vertex of degree at most  $k$ . Our contribution is an algorithm computing a maximum clique for this class of graphs in time  $\mathcal{O}((n-k)f(k))$ , where  $f(k)$  is the time complexity of computing a maximum clique in a graph of order  $k$ , thus improving the previous best result. We also prove some structural properties for inductive  $k$ -independent graphs.

## 1 Introduction and notations

The main contribution of this work is an algorithmic approach for cliques in inductive  $k$ -independent graphs and  $k$ -degenerate graphs.

Inductive  $k$ -independent graphs, introduced in Akcoglu et al. [1], are a natural generalisation of chordal graphs and have been studied more extensively in Borodin et al. [2]. In these two papers, the authors studied algorithmic and structural properties of this new family of graphs, they showed that the general recognition problem is NP-complete but several natural classes of graphs are inductive  $k$ -independent for a small constant  $k$ . For instance, chordal graphs are inductive 1-independent, planar graphs are inductive 3-independent and claw-free graphs are inductive 2-independent. This property is not edge-wise monotone as for instance complete graphs and stars graphs are inductive 1-independent. In the second paper [2], the authors also studied  $k$ -approximation algorithms for the maximum independent set problem and minimum vertex colouring problem. For the minimum vertex cover problem they showed that one can achieve a  $(2 - \frac{1}{k})$ -approximation. Here we will also consider degenerate graphs [3], a subclass of inductive  $k$ -independent ones. Roughly speaking, degeneracy is a common measure of the sparseness of a graph. Many real-life graphs are sparse, have low degeneracy, and follow a power-law degree distribution [4]. A graph will be said to be  $k$ -degenerate if every induced subgraph has a vertex of degree at most  $k$ .

The CLIQUE problem asks, given a graph  $G$  and an integer  $k$ , whether  $G$  contains a clique on  $k$  vertices. The MAXIMUM CLIQUE problem is to find a clique of maximum cardinality.

Formally, let  $G = (V, E)$  be a graph of  $n$  vertices and  $m$  edges. The graph  $\overline{G}$  is its complement. If  $X \subset V$ , the *subgraph* of  $G$  induced by  $X$  is denoted by  $G[X]$ . The set  $N_G(x)$  is called the *open neighbourhood* of the vertex  $x$  in  $G$ , that is the set of neighbours of  $x$  in the graph. The *closed neighbourhood* of  $x$  is defined as  $N_G[x] = N_G(x) \cup x$ . When the context is clear we write  $N(x)$  and  $N[x]$  instead of  $N_G(x)$  and  $N_G[x]$ . The order of a maximum independent set in  $G$  is denoted by  $\alpha(G)$ , for a maximum clique it is denoted  $\omega(G)$ . Given an ordering  $v_1, \dots, v_n$  of the vertices of  $G$  we denote by  $V_i$  the set of vertices following  $v_i$  including itself in this order, that is, the set  $\{v_i, \dots, v_n\}$ . Let  $G_i^+$  denote the induced subgraph  $G[N[v_i] \cap V_i]$  and  $G_i$  the induced subgraph  $G[N(v_i) \cap V_i]$ . By  $\Delta(G)$  we will denote the *maximum degree* of a graph. For a given ordering  $\sigma$  of the vertices of a graph  $G$ , let  $\Delta_i$  be the maximum degree of  $G_i$  and let  $\Delta_\sigma = \max_i(\Delta_i)$ .  $\overline{\Delta}_i$  is the maximum degree of  $\overline{G}_i$  and  $\overline{\Delta}_\sigma = \max_i(\overline{\Delta}_i)$ . An *inductive  $k$ -independence ordering* is an ordering of vertices  $v_1, v_2, \dots, v_n$  such that for any  $v_i$ ,  $1 \leq i \leq n$ ,  $\alpha(G[N(v_i) \cap V_i]) \leq k$ . The minimum of such  $k$  over all orderings is called the inductive independence number of  $G$ , which we denote as  $\lambda(G)$ . A graph  $G$  is *inductive  $k$ -independent* if  $\lambda(G) \leq k$ . A graph is  *$k$ -degenerate* if there is an ordering  $v_1, \dots, v_n$  of its vertices such that  $v_i$ ,  $1 \leq i \leq n$ ,  $|N(v_i) \cap V_i| \leq k$ . The approximation ratio of an algorithm for a maximization problem is the ratio  $\frac{\text{size of the optimum solution}}{\text{size of the algorithm solution}}$ .

The paper is organised as follows. In Section 2 we establish some properties of cliques in general graphs with any orderings of vertices. In Section 3 we present the fixed-parameter and approximation algorithms for inductive  $k$ -independent graphs. In Section 4 we prove the correctness of the algorithm finding a maximum clique in  $k$ -degenerate graphs. Finally, in Section 5 we will present some miscellaneous results for inductive  $k$ -independent graphs.

## 2 General results for cliques in graphs

In this section we give some general results concerning maximal cliques related to orderings of vertices in general graphs.

**Lemma 1** *Let  $G$  be a graph and let  $v_1, \dots, v_n$  be any ordering of its vertices. Then every maximal clique of  $G$  belongs to exactly one subgraph  $G_i^+$ .*

**Proof:** Let  $K$  be any maximal clique of  $G$ . Let  $V(K)$  denote the set of its vertices. Let  $x \in V(K)$  be the vertex of  $K$  with the smaller ranking in the ordering. Observe that  $V(K) \subseteq V(G_x^+)$ . Assume by contradiction that there exists  $y \in V(G)$  and  $y \neq x$  such that  $V(K) \subseteq V(G_y^+)$ . By the maximality property of  $K$ , observe that necessarily  $y \in V(K)$ . By hypothesis,  $y$  has higher ranking than  $x$  in the ordering. This implies that  $x \notin V(G_y^+)$ . Since we have that  $x \in V(K)$  and  $V(K) \subseteq V(G_y^+)$ , this gives the contradiction.  $\square$

**Lemma 2** *Let  $G$  be a graph. If  $G$  has an ordering of its vertices such that for each  $i$ , a maximum clique of  $G_i^+$  can be found in time  $\mathcal{O}(T_i)$ , then a maximum clique of  $G$  can be found in time  $\mathcal{O}(\sum_{i=1}^n T_i)$ .*

**Proof:** Let  $G$  be a graph. Suppose it has an ordering of its vertices  $v_1, \dots, v_n$  such that, for all  $i$ , a maximum clique of  $G_i^+$  can be found in time  $\mathcal{O}(T_i)$ . For each vertex of the order compute a maximum clique in its neighbourhood. This can be done in time  $\mathcal{O}(\sum_{i=1}^n T_i)$ . Then, among these cliques return the largest one. By Lemma 1, this clique is a maximum one of  $G$  and the algorithm runs in time  $\mathcal{O}(\sum_{i=1}^n T_i)$ . □

**Lemma 3** *Let  $G$  be a graph and  $\beta_1, \dots, \beta_n \in \mathbb{R}$ . Let  $\beta = \max_i(\beta_i)$ . Assume that  $G$  has an ordering of its vertices such that for all  $i$ , there is a  $\beta_i$ -approximation algorithm for the maximum clique of  $G_i^+$  running in time  $\mathcal{O}(T_i)$ . Then there is a  $\beta$ -approximation algorithm for the maximum clique of  $G$  running in time  $\mathcal{O}(\sum_{k=1}^n (T_k))$ .*

**Proof:** For every graph  $G_i^+$  compute a  $\beta_i$ -approximation of its maximum clique  $K_i$ , call it  $K'_i$ . This will take  $\mathcal{O}(\sum_{k=1}^n (T_k))$  time. Assume that, w.l.o.g, a largest such clique appears in some graph  $G_j^+$ . Denote by  $OPT_i$  the size of the maximum clique of  $G_i^+$  and by  $OPT$  the size of the maximum clique of  $G$ .

By Lemma 1 there exists  $r \in \{1, \dots, n\}$  such that  $OPT_r = OPT$ . Therefore

$$|V(K'_j)| = \max_i |V(K'_i)| \geq \frac{OPT_r}{\beta_r} \geq \frac{OPT}{\max_i \beta_i}$$

□

**Corollary 1** *Let  $G$  be a graph. Assume  $G$  has an ordering of its vertices such that for all  $i$ , there is an  $\beta$ -approximation algorithm for the maximum clique of  $G_i^+$  running in time  $\mathcal{O}(T)$ . Then there is an  $\beta$ -approximation algorithm for the maximum clique of  $G$  running in time  $\mathcal{O}(nT)$ .*

### 3 Algorithms for cliques for inductive $k$ -independent graphs

Using the general results of Section 2 in connection with Theorem 2 below, we prove an approximation algorithm of ratio  $\Delta(\overline{G})/\log(\log(\Delta(\overline{G}))(k+1))$ . (see Corollary 4).

**Theorem 2** [5] *Given a  $K_k$ -free graph  $G$ , there is a polynomial time algorithm that achieves a  $\Delta(G)/\log(\log(\Delta(G))/(k))$  approximation for maximum independent set.*

**Theorem 3** *Given an inductive  $k$ -independent graph and a  $k$ -inductive ordering of its vertices, there is a polynomial time algorithm that achieves a  $\overline{\Delta}_\sigma/\log(\log(\overline{\Delta}_\sigma))/(k+1)$  approximation for maximum clique.*

**Proof:** By definition we have that for each  $i$ ,  $\alpha(G_i^+) \leq k$ . Therefore every graph  $\overline{G}_i$  is  $(k+1)$ -clique free. By Theorem 2 we can approximate the maximum independent set of every  $\overline{G}_i^+$  with ratio  $\overline{\Delta}_\sigma/\log(\log(\overline{\Delta}_\sigma))/(k+1)$  in polynomial time. Thus, this gives an approximation for the maximum clique of every  $G_i^+$  with same ratio. Now by Corollary 1 we have the claimed result.  $\square$

**Corollary 4** *Given an inductive  $k$ -independent graph and a  $k$ -inductive ordering of its vertices, there is a polynomial time algorithm that achieves a  $\Delta(\overline{G})/\log(\log(\Delta(\overline{G}))/(k+1))$  approximation for the maximum clique.*

**Proof:** Straightforward from the fact that  $\overline{\Delta}_\sigma \leq \Delta(\overline{G})$ .  $\square$

Observe that the complement of a  $K_k$ -free graph is inductive  $k$ -independent. Thus if we could achieve a better approximation ratio than the one proved in Corollary 4, we would also improve Theorem 2. Now for inductive 2-independent graphs we can achieve a better approximation ratio thanks to Theorem 5.

**Theorem 5** [6] *Given a  $K_3$ -free graph  $G$ , there is a polynomial time algorithm that achieves a  $(\Delta(G)-1)^2/(\Delta(G)\ln(\Delta(G))-\Delta(G)+1)$  approximation for the maximum independent set.*

**Theorem 6** *Given an inductive 2-independent graph and a 2-inductive ordering of its vertices, there is a polynomial time algorithm that achieves a  $(\Delta(\overline{G})-1)^2/(\Delta(\overline{G})\ln(\Delta(\overline{G}))-\Delta(\overline{G})+1)$  approximation for the maximum clique.*

**Proof:** Same as for Theorem 2.  $\square$

Now we show that the decision version of the CLIQUE problem is fixed-parameter tractable when parametrized by the size  $p$  of the clique we seek for, for inductive  $k$ -independent graphs:

**Theorem 7** *Given an inductive  $k$ -independent graph and a  $k$ -inductive ordering of its vertices, there is a fixed-parameter algorithm solving the CLIQUE problem in  $\mathcal{O}(p^2(p+k-1)^{kp}n)$  time where  $p$  is the size of the clique.*

**Proof:** Ramsey's theorem states that for any two positive integers  $i, c$  there exists a positive integer  $R(i, c)$  such that any graph with at least  $R(i, c)$  vertices contains either an independent set on  $i$  vertices or a clique on  $c$  vertices or both.

It is shown [7, p. 65] that  $R(i, c) \leq \binom{i+c-2}{c-1}$ . Now with  $i = k + 1$  and  $c = p$  then if a graph with at least  $\binom{p+k-1}{p-1} = \binom{p+k-1}{k} \leq (p+k-1)^k$  vertices does not contain an independent set of size  $k+1$ , then it must have a clique on  $p$  vertices. Let  $G$  be our inductive  $k$ -independent graph and let  $v_1, \dots, v_n$  be its  $k$ -independence ordering. We know by definition that for each  $i$ ,  $\alpha(G_i^+) < k+1$ . Therefore, if for some  $i$ ,  $|V(G_i^+)| \geq (p+k-1)^k$  then  $G_i^+$  contains a clique on  $p$  vertices by Ramsey's theorem. This can be checked in time  $\mathcal{O}(n)$ . Now suppose that for each  $i$ ,  $|V(G_i^+)| < (p+k-1)^k$ . We iterate over the vertices in the  $k$ -independence order and check for every vertex if the graph  $G_i^+$  contains a clique of size  $p$ . By Lemma 1, this procedure will return a clique of size  $p$ , if any, in  $G$ . To achieve this, we generate all subsets of size  $p$  in time  $\mathcal{O}\left(\binom{p+k-1}{p}^k\right) = \mathcal{O}((p+k-1)^{kp})$  and check if all the  $\mathcal{O}(p^2)$  edges are present, which can be done in  $\mathcal{O}(p^2)$  with an adjacency matrix. This takes  $\mathcal{O}(p^2(p+k-1)^{kp})$  in total. Thus by Lemma 2 we can conclude that the algorithm runs in time  $\mathcal{O}(p^2(p+k-1)^{kp}n)$   $\square$

## 4 An algorithm for the MAXIMUM CLIQUE problem for $k$ -degenerate graphs

To our knowledge, the best known algorithm to compute the maximum clique of a  $k$ -degenerate graph  $G$  has been proved in Buchanan et al. [8] with complexity  $\mathcal{O}(nm + n2^{k/4})$ . The exponential factor is the time needed to compute a maximum clique in a graph of order  $k$  and is taken from the literature. We do not improve that exponential dependence. We get rid of the quadratic term and reduce the linear term to  $(n-k)$ . This gives a time complexity of  $\mathcal{O}((n-k)f(k))$  where  $f(k)$  is the time complexity to find a maximum clique in a graph of order  $k$ . To achieve this, using a degeneracy ordering, we first construct an algorithm to compute  $n-k+1$  induced subgraphs:  $G_i^+$  for  $i = 1, \dots, n-k$  and the graph  $G_{n-k+1}^*$  which is the subgraph induced on the last  $k$  vertices of the ordering. Then we show how to find the maximum clique of  $G$ .

**Lemma 4** *Given a  $k$ -degenerate graph  $G$ , there is an algorithm constructing the induced subgraphs  $G_i$  for  $i = 1, \dots, (n-k)$  and the graph  $G_{n-k+1}^*$  in  $\mathcal{O}((n-k+1)k^2 \log(k))$  time, using  $\mathcal{O}(m)$  memory space.*

**Proof:** Assume that  $G$  is represented by its adjacency lists, using therefore  $\mathcal{O}(m)$  memory space. Degeneracy, along with a degeneracy ordering, can be computed by greedily removing a vertex with smallest degree (and its edges) from the graph until it is empty. The degeneracy ordering is the order in which vertices are removed from the graph and this algorithm can be implemented in  $\mathcal{O}(m)$  time [9].

Using this degeneracy ordering we construct below the vertex sets of the graphs  $G_i$  for  $i = 1, \dots, (n-k)$  and of the graph  $G_{n-k+1}^*$  as follows. Assume that initially all the vertices of  $G$  are coloured blue. Consider iteratively, one by one, the first  $n-k$  vertices  $v_1, v_2, \dots, v_{n-k}$  of the ordering. At Step  $i$ , we start by

colouring vertex  $v_i$  red. Then, we scan its neighbourhood (using an adjacency list), we skip its red neighbours and put its blue neighbours in  $V(G_i)$ . This is because if one of its neighbour is red, it means that it appears before it in the ordering and thus should not be put in  $V(G_i)$ . At the end the  $(n - k)$  first iterations put the remaining  $k$  vertices in the vertex set  $V(G_{n-k+1}^*)$ . This construction can be done in  $\mathcal{O}(m)$  time since each iteration takes time proportional to the degree of the vertex we are considering in the order.

Now we construct the edge sets of the graphs  $G_i$  for  $i = 1, \dots, n - k$  and of the graph  $G_{n-k+1}^*$  as follows. For the vertex sets  $V(G_i)$  for  $i = 1, \dots, (n - k)$  and for  $V(G_{n-k+1}^*)$  we start by sorting their vertices following the degeneracy ordering in time  $\mathcal{O}(k \log(k))$  for each such set. This takes total time  $\mathcal{O}((n - k + 1)k \log(k))$ . This will give us, for each vertex  $v_1, \dots, v_n$ , a sorted array  $D_i = d_1, \dots, d_k$  containing its neighbours coming later in the degeneracy ordering. This takes space  $\mathcal{O}(m)$ : every edge  $(i, j)$  appears at most one time (in the list  $d_i$  if  $i < j$  for instance). Using this structure, we now show how to build the edge sets. Assume that we want to build the edge set of the graph  $G_i$ . For each element  $d_j$  for  $j = 1, \dots, k$  of  $D_i$ , we check for every element  $d_{j'}$  of  $D_i$  with  $j' > j$  if it appears in  $D_{d_j}$ . If it is the case, we add the corresponding edge. This is done in  $\mathcal{O}(k^2 \log(k))$  for all the elements of  $D_i$ . Therefore, to build all the graphs  $(G_i)$  for  $i = 1, \dots, n - k$  and of the graph  $G_{n-k+1}^*$  we need, overall,  $\mathcal{O}((n - k + 1)k^2 \log(k) + m) = \mathcal{O}((n - k + 1)k^2 \log(k) + nk) = \mathcal{O}((n - k + 1)k^2 \log(k))$ , and  $\mathcal{O}(m)$  space, as claimed.  $\square$

**Theorem 8** *Given a  $k$ -degenerate graph  $G$ , there is an algorithm finding a maximum clique in  $\mathcal{O}(f(k)(n - k))$  time, where  $f(k)$  is the time complexity of finding a maximum clique in a graph of order  $k$ .*

**Proof:** Using Lemma 4, we start by constructing the subgraphs  $G_i$  for  $i = 1, \dots, (n - k)$  and  $G_{n-k+1}^*$  in time  $\mathcal{O}((n - k + 1)k^3)$ . Then for each of these graphs, since they are of size  $k$ , we find their maximum clique  $K_i$  in time  $f(k)$ . Among these computed cliques find the one of largest size, call it  $C$ . If  $C$  is an induced subgraph of some graph  $G_i$  for  $i$  in  $\{1, \dots, (n - k - 1)\}$ , then, by Lemma 1,  $G[V(C) \cup \{v_i\}]$  is a maximum clique of  $G$  and we output it. In the other case, assume that  $C$  is an induced subgraph of  $G_{n-k+1}^*$ . There are two cases. First case there is no other maximum clique  $K_j$ , of same largest size, and which is an induced subgraph of a  $G_j$  for  $j$  in  $\{1, \dots, (n - k)\}$ . Then we simply output  $C$  using again Lemma 1. In the second case, assume that there is a maximum clique  $K_j$  of same largest size induced in some graph  $G_j$  for a  $j$  in  $\{1, \dots, (n - k)\}$ . We have that  $G[V(K_j) \cup \{v_j\}]$  is a maximum clique of  $G$  (using again Lemma 1) and such that  $|V(K_j) \cup \{v_j\}| = |V(C)| + 1$ . Therefore in this case we output  $G[V(K_j) \cap \{v_j\}]$ .

Since the graphs  $G_i^+$  for  $i$  in  $\{n - k + 1, \dots, n\}$  are induced subgraphs of  $G_{n-k+1}^*$ , this procedure will output a correct maximum clique of the graph. Overall this will take time  $\mathcal{O}(f(k)(n - k + 1) + k^3(n - k + 1)) = \mathcal{O}(f(k)(n - k + 1))$ . Now we modify the algorithm to get the claimed complexity. Consider  $G_{n-k+1}^*$ .

If it is of size  $k$ , then  $G_{n-k+1}^*$  and  $G_{n-k}$  are the same graphs. In this case we need overall time  $\mathcal{O}(f(k)(n-k))$ . Conversely if  $|V(G_{n-k+1}^*)| < k$  then the algorithm previously described requires  $\mathcal{O}(f(k)(n-k))$  time for graphs  $G_i$  for  $i \in \{1, \dots, n-k\}$ , and  $\mathcal{O}(f(k-1))$  time for  $G_{n-k+1}^*$ . Thus overall it needs  $\mathcal{O}(f(k-1)) + \mathcal{O}(f(k)(n-k)) = \mathcal{O}(f(k)(n-k))$  time, as claimed.  $\square$

## 5 Miscellaneous results

In this section we give some results of general interest, not related to algorithms for cliques. We start by proving that any induced subgraph of an inductive  $k$ -independent graph is inductive  $k$ -independent. This result was proved in the original paper [2] but there was a mistake in the proof. The authors assumed that if a graph  $H$  is not inductive  $k$ -independent then for any vertex  $v \in V(H)$ ,  $\alpha(N(v)) > k$ , which is incorrect. This is the reason we include a complete proof here.

**Lemma 5** [2] *Any induced subgraph of an inductive  $k$ -independent graph is an inductive  $k$ -independent graph.*

**Proof:** Assume  $G = (V, E)$  is an inductive  $k$ -independent graph and let  $\sigma$  be the related ordering. Consider a subset  $V' \subset V$  and let  $G' = G[V']$ .

For every  $i$ , the set  $N_{G'}[v_i] \cap V_i$  is a subset of  $N_G[v_i] \cap V_i$ . Then  $G'_i$  is an induced subgraph of  $G_i$  and  $\alpha(G'_i) \leq k$ . Therefore taking the vertices of  $V'$  in the same order as they appear in  $\sigma$  yields an inductive  $k$ -independence ordering for  $G'$ .  $\square$

**Theorem 9** [10] *Let  $G$  be a  $k$ -connected graph with  $n \geq 3$  vertices. If  $\alpha(G) \leq k$ , then  $G$  is Hamiltonian.*

**Proposition 10** *Let  $G$  be an inductive 2-independent graph and  $\sigma = v_1, \dots, v_n$  a 2-inductive ordering of its vertices. For every  $i$ , either the induced subgraph  $G_i^+$  is Hamiltonian or  $G_i^+$  is covered by at most two cliques.*

**Proof:** If for some  $j$  the graph  $G_j^+$  has strictly less than 3 vertices then observe that  $G_j^+$  can be covered by at most two cliques and the theorem is true. Therefore consider now only the induced subgraphs with more than three vertices. Let  $G_i^+$  be such an induced subgraph. We have two cases:

1.  $G_i^+$  is  $k$ -connected with  $k \geq 2$ . Since by definition  $\alpha(G_i^+) \leq 2$  then by Theorem 10,  $G_i^+$  is Hamiltonian.
2.  $G_i^+$  is  $k$ -connected with  $k = 1$ . By definition we know that  $v_i$  is adjacent to every vertex in  $V(G_i)$ . Therefore removing a vertex in  $V(G_i)$  can not disconnect the graph. Thus, since the graph  $G_i^+$  is 1-connected and no vertex in  $V(G_i)$  can be a cut vertex then necessarily  $v_i$  is a cut vertex.

This implies that the graph  $G_i$  is disconnected and that it has at most two connected components (otherwise it would have an independent set of size greater than two). Call these components  $G_{i1}$  and  $G_{i2}$ . Since both of these graphs can have an independent set of size at most one, (otherwise  $G_i$  would have an independent set of size greater than two) then  $G_{i1}$  and  $G_{i2}$  are cliques. Thus the theorem holds in that case.

□

## 6 Acknowledgement

I am grateful to professor Michel Habib for all his valuable suggestions and comments.

## References

- [1] K. Akcoglu, J. Aspnes, B. DasGupta, M.-Y. Kao, Opportunity cost algorithms for combinatorial auctions, in: Computational Methods in Decision-Making, Economics and Finance, Springer, 2002, pp. 455–479.
- [2] Y. Ye, A. Borodin, Elimination graphs, ACM Trans. Algorithms 8 (2) (2012) 14:1–14:23.
- [3] D. R. Lick, A. T. White,  $d$ -degenerate graphs, Canad. J. Math. 22 (1970) 1082–1096.
- [4] F. Chung, Graph theory in the information age, Notices of the AMS 57 (6) (2010) 726–732.
- [5] M. M. Halldórsson, J. Radhakrishnan, Improved approximations of independent sets in bounded-degree graphs via subgraph removal, Nordic J. of Computing 1 (4) (1994) 475–492.
- [6] J. B. Shearer, A note on the independence number of triangle-free graphs, ii, J. Comb. Theory, Ser. B 53 (2) (1991) 300–307.
- [7] S. Jukna, Extremal combinatorics - with applications in computer science, Texts in theoretical computer science, Springer, 2001.
- [8] A. Buchanan, J. L. Walteros, S. Butenko, P. M. Pardalos, Solving maximum clique in sparse graphs: an  $\mathcal{O}(nm + n2^{d/4})$  algorithm for  $d$ -degenerate graphs, Optimization Letters 8 (5) (2014) 1611–1617.
- [9] V. Batagelj, M. Zaversnik, An  $\mathcal{O}(m)$  algorithm for cores decomposition of networks, CoRR cs.DS/0310049.
- [10] V. Chvátal, P. Erdős, A note on hamiltonian circuits, Discrete Mathematics 2 (2) (1972) 111 – 113.